FPT Education

FPT POLYTECHNIC

LẬP TRÌNH ANDROID 2

FRAGMENT

www.poly.edu.vn

GIỚI THIỆU BÀI HỌC

- Tổng quan Fragment
- Sử dụng Fragment
- Giao tiếp giữa Activity và Fragment
- Sử dụng ViewPager2 và TabLayout

Mục tiêu

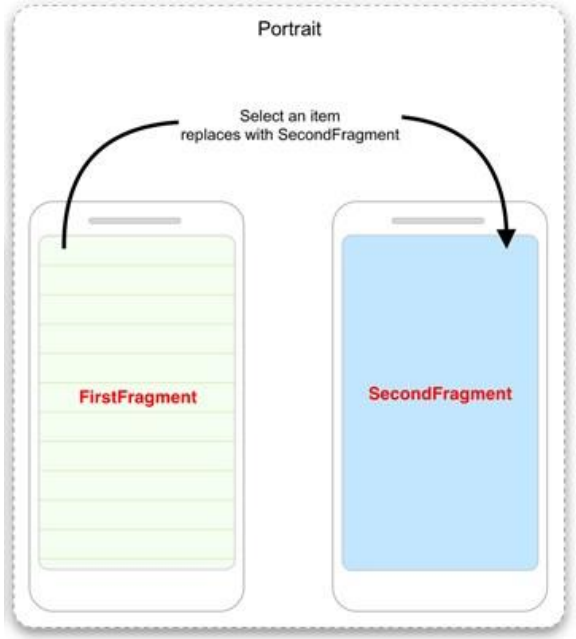
- Tổng quan Fragment
- Sử dụng Fragment
- Giao tiếp giữa Activity và Fragment
- Sử dụng ViewPager2 và TabLayout

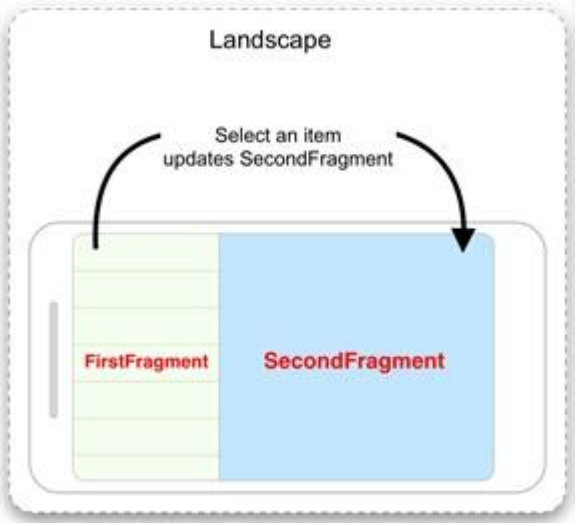
FRAGMENT

…

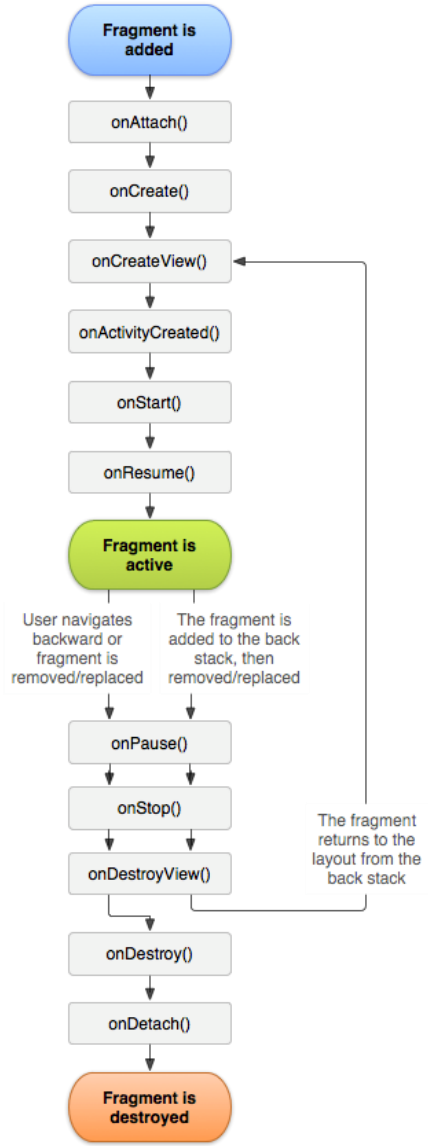
TỔNG QUAN VỀ FRAGMENT

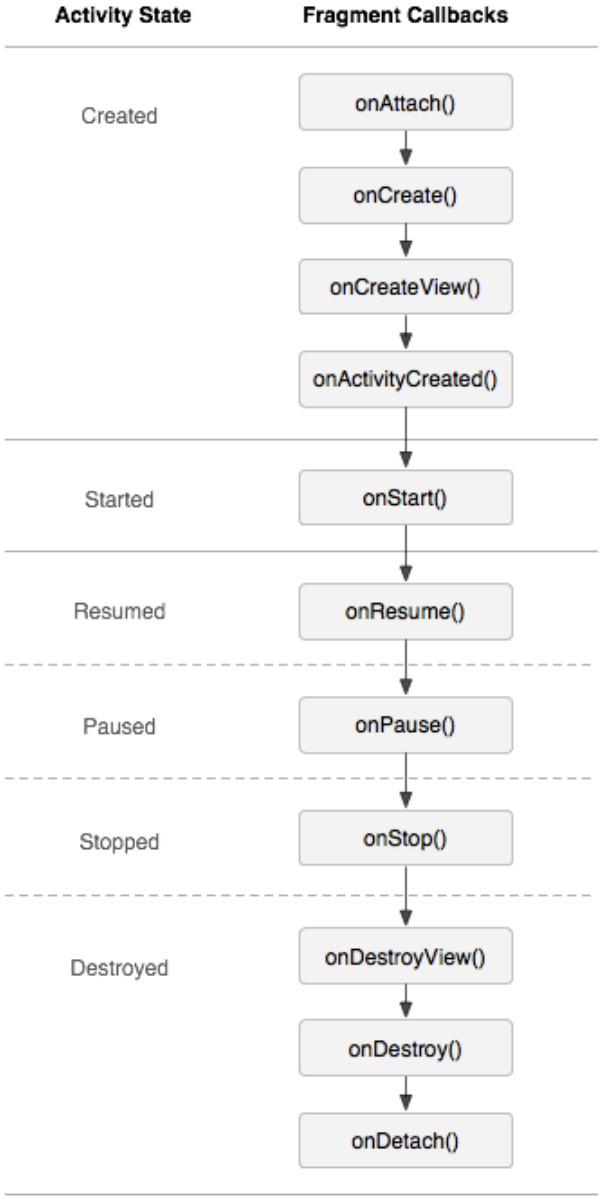
- **Fragment** là một phân đoạn giao diện được nhúng vào một Activity, fragment không chạy độc lập với activity được.
- Fragment giúp chương trình linh động và hướng module hơn.
- Một fragment có thể chiếm một phần hoặc toàn bộ giao diện Activity. Một Activity có thể có nhiều fragment.
- Fragment có thể cố định hoặc có thể được tạo, thêm, xóa,… trong quá trình Activity hoạt động





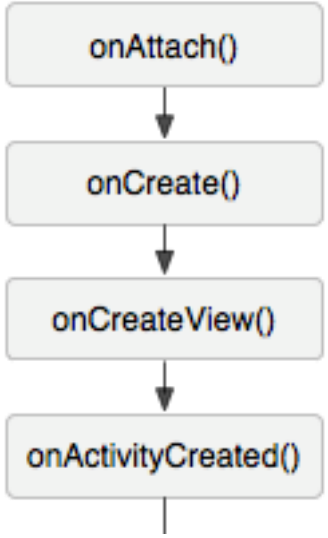
VÒNG ĐỜI FRAGMENT





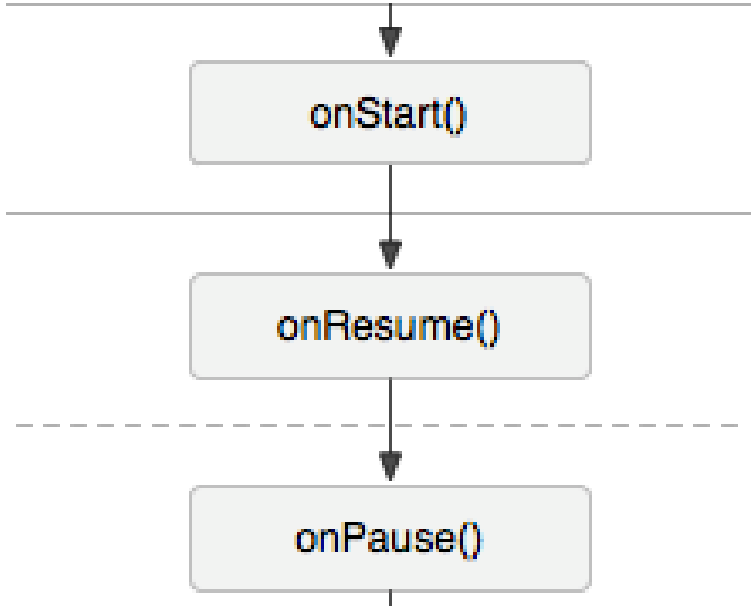
VÒNG ĐỜI FRAGMENT

- **onAttach():** Được gọi khi fragment đã được liên kết với một activity
- **onCreate():** Được sử dụng để khởi tạo fragment.
- **onCreateView():** Được sử dụng để tạo fragment view
- **onActivityCreated():** Được gọi khi fragment và view của fragment được khởi tạo, dùng để hoàn thành nốt công đoạn khởi tạo fragment và activity



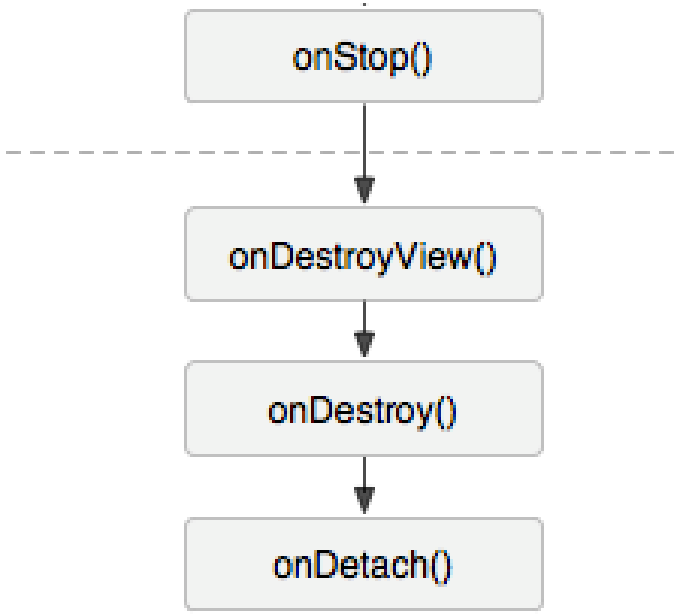
VÒNG ĐỜI FRAGMENT

- **onStart():** Được gọi để hiển thị fragment.
- **onResume():** Khi fragment hiển thị và có thể tương tác
- **onPause():** Được gọi khi fragment không còn hiển thị và người dùng đang rời khỏi fragment.



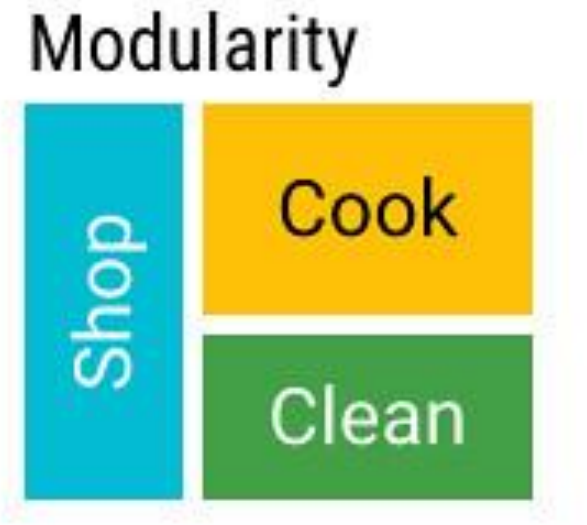
VÒNG ĐỜI FRAGMENT

- **onStop():** Được gọi để dừng fragment.
- **onDestoryView():** Giao diện view fragment bị xóa sau khi thực hiện.
- **onDestroy():** Được gọi khi hủy fragment.
- **onDetach():** Được gọi ngay sau khi fragment bị tách ra khỏi activity.

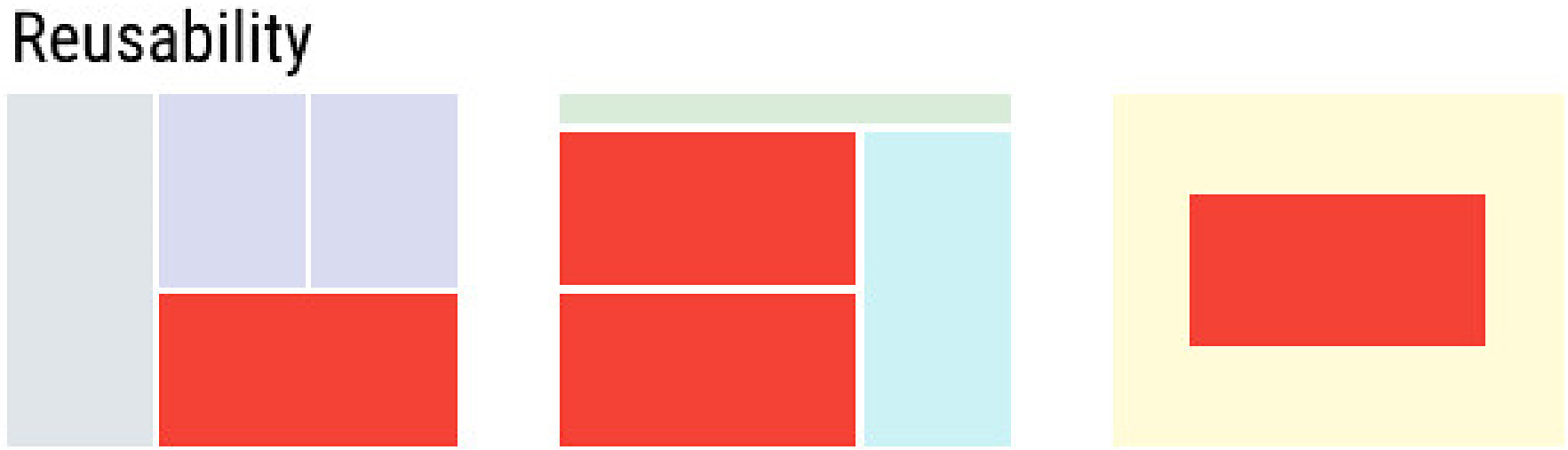


ƯU ĐIỂM CỦA FRAGMENT

- **Module hóa (Modularity):** Với các Activity phức tạp thì code sẽ được implement ở các Fragment. Mỗi Fragment là một module độc lập. Điều này sẽ làm cho code dễ tổ chức và bảo trì tốt hơn.



- **Tái sử dụng (Reusability):** Viết code implement các tương tác với người dùng hoặc các phần UI vào fragment để có thể chia sẻ chúng với các Activity khác.



ƯU ĐIỂM CỦA FRAGMENT

- **Hỗ trợ đa màn hình (Adaptability):** Fragment cung cấp cách để trình bày giao diện người dùng (UI) phù hợp và tối ưu cho các loại thiết bị Android có kích thước màn hình và mật độ điểm ảnh khác nhau.

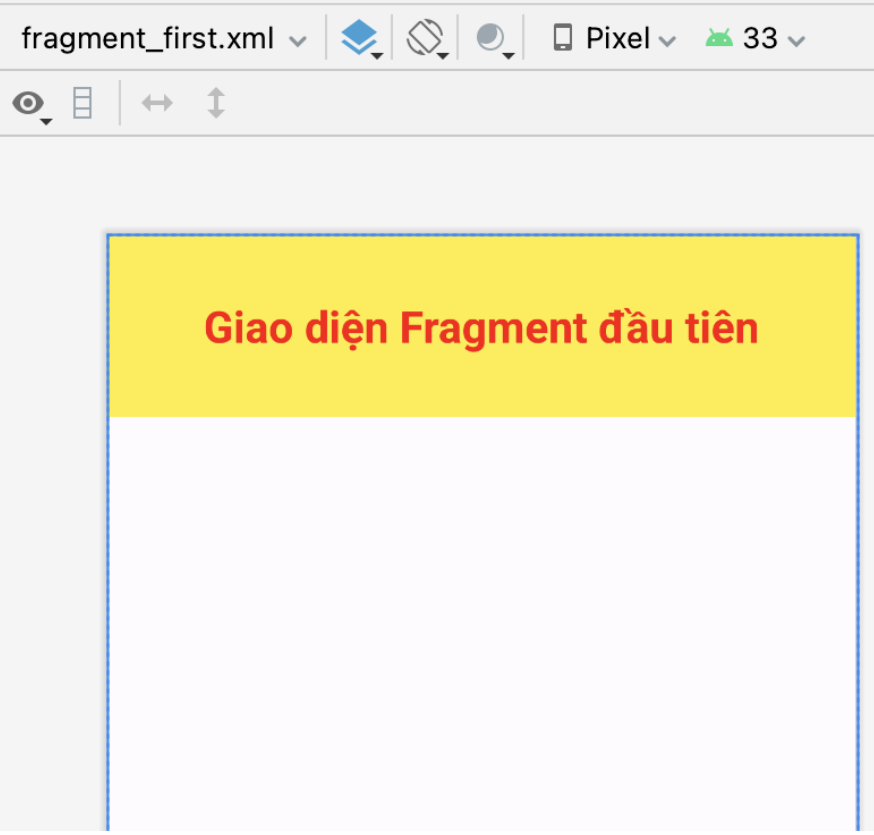
SỬ DỤNG FRAGMENT

...

❑ **Bước 1:** Tạo giao diện cho Fragment

fragment_first.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="100dp"
        android:text="Giao diện Fragment đầu tiên"
        android:textSize="24sp"
        android:gravity="center"
        android:textStyle="bold"
        android:textColor="#FF0000"
        android:background="#FFEB3B"/>

</LinearLayout>
```



TẠO MỘT FRAGMENT

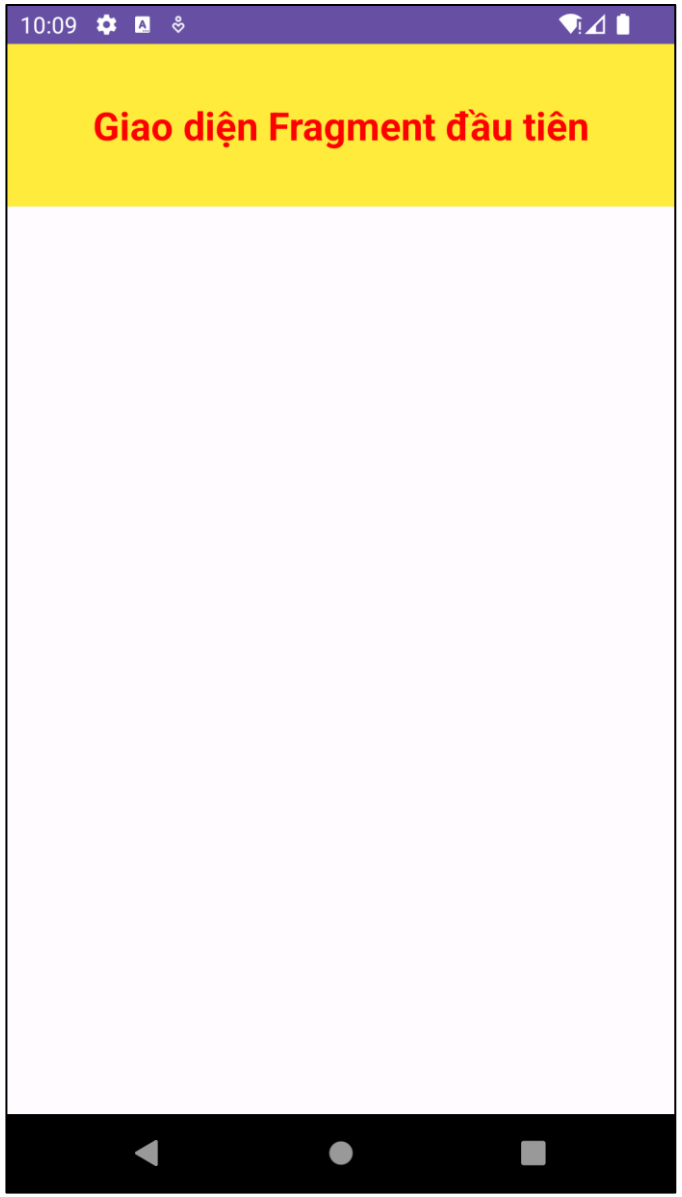
- **Bước 2:** Tạo một class mới kế thừa từ Fragment và gán layout tạo ở bước 1

```
public class FirstFragment extends Fragment {

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_first, container,  attachToRoot: false);
        return view;
    }
}
```

TẠO MỘT FRAGMENT

- **Bước 3:** Đưa Fragment lên Activity
  - **Cách 1:** Sử dụng XML

```
<fragment
    android:id="@+id/firstFragment"
    class="com.dinhnt.android02.FirstFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
```



- **Bước 3:** Đưa Fragment lên Activity
  - **Cách 2:** Có thể sử dụng **FragmentManager** để quản lý các Fragment trong khi đang chạy chương trình, Các bước như sau:
    - Dùng FragmentManager.beginTransaction() để lấy 1 instance của FragmentTransaction.
    - Khởi tạo Fragment.
    - Dùng FragmentTransaction.add() để thêm fragment vào.
    - Gọi FragmentTransaction.commit() để chấp nhận.

- **Bước 3:** Đưa Fragment lên Activity
  - **Cách 2:** Có thể sử dụng **FragmentManager** để quản lý các Fragment trong khi đang chạy chương trình
    - Tạo layout để hiển thị Fragment

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/andr
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <RelativeLayout
        android:id="@+id/relativeLayout"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>

</LinearLayout>
```

- **Bước 3:** Đưa Fragment lên Activity
  - **Cách 2:** Có thể sử dụng **FragmentManager** để quản lý các Fragment trong khi đang chạy chương trình
    - Gọi ***FragmentManager*** để thêm một Fragment lên RelativeLayout đã tạo

```
FragmentManager fragmentManager = getSupportFragmentManager();
FirstFragment fragment = new FirstFragment();
fragmentManager
        .beginTransaction()
        .add(R.id.relativeLayout, fragment)
        .commit();
```

TẠO MỘT FRAGMENT

- Ngoài ra có thể dùng thêm các lệnh của FragmentTransaction như : **add()**, **remove()**, **replace()**, **hide()**, **show()**, **detach()** (api 13), **attach()** (api 13).

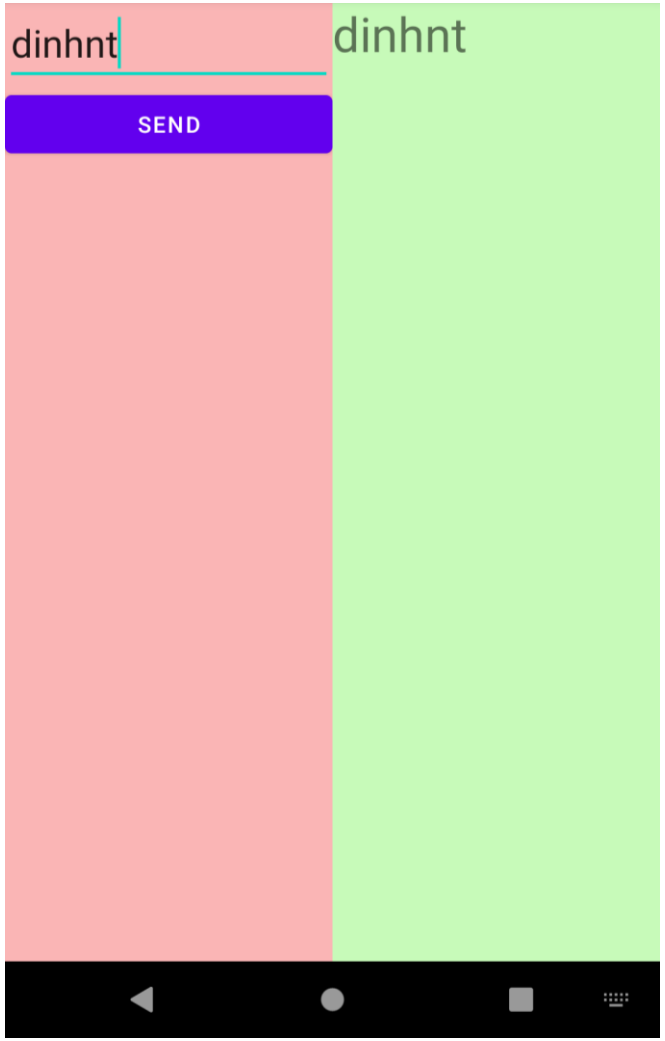
```
FragmentManager fragmentManager = getSupportFragmentManager();
fragmentManager
        .beginTransaction()
        .remove(fragment1)
        .add(R.id.relativeLayout, fragment2)
        .show(fragment3)
        .hide(fragment4)
        .replace(R.id.relativeLayout, fragment5)
        .commit();
```

GIAO TIẾP GIỮA ACTIVITY VÀ FRAGMENT

- Một fragment gắn liền với activity chứa nó. Activity có thể gọi trực tiếp các phương thức public của Fragment thông qua tham chiếu đến đối tượng Fragment.
- Nếu ta không giữ lại các tham chiếu đến Fragment khi tạo nó ta cũng có thể tìm nó thông qua FragmentManager, dùng **findFragmentById()** hoặc **findFragmentByTag()**.
- Fragment cũng có thể truy cập tới Activity chứa nó bằng phương thức **Fragment.getActivity()**.

GIAO TIẾP GIỮA ACTIVITY VÀ FRAGMENT

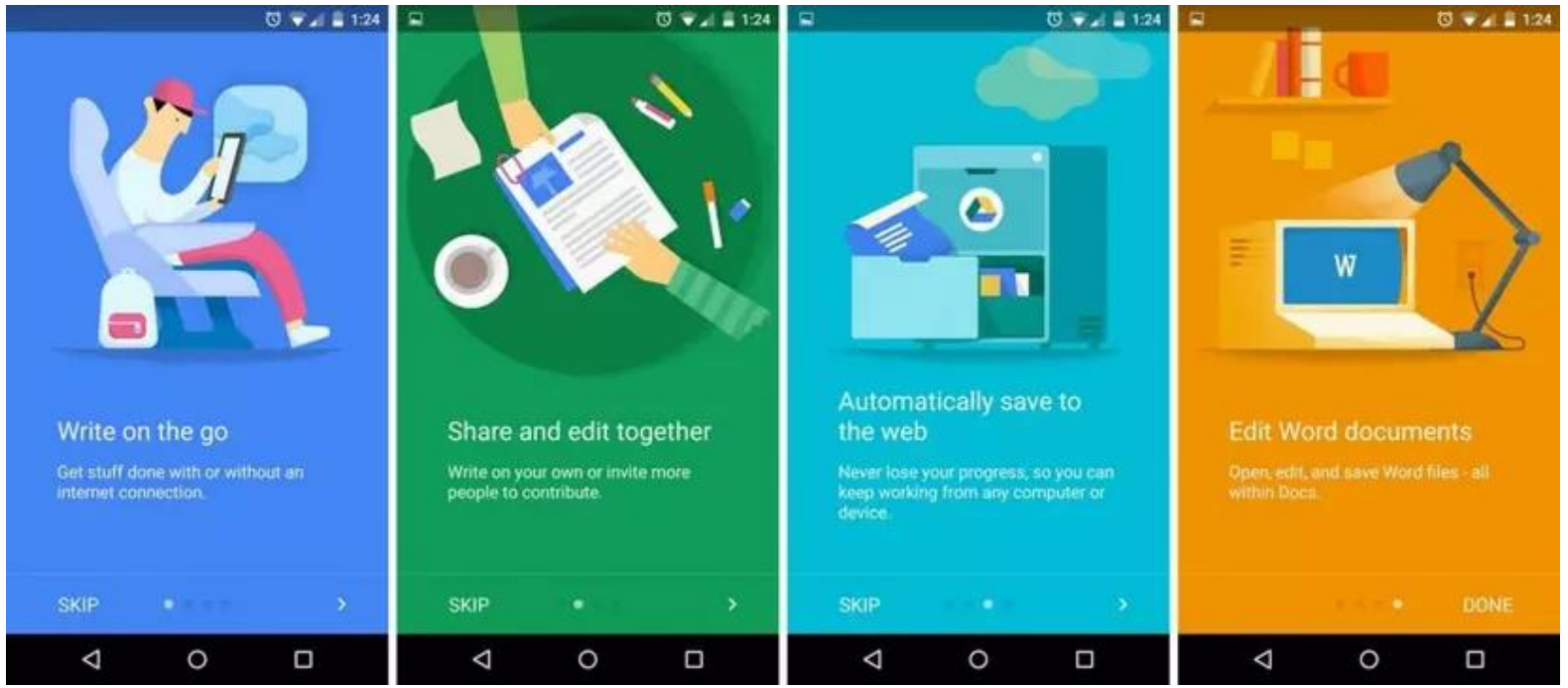
```
btnSend.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String data = edtInput.getText().toString();
        FragmentManager fragmentManager = requireActivity().getSupportFragmentManager();
        FragmentRightBai02 fragmentRightBai02 = (FragmentRightBai02) fragmentManager
                                                .findFragmentById(R.id.fragRightBai02);
        fragmentRightBai02.txtResult.setText(data);
    }
});
```



TABLAYOUT VÀ VIEWPAGER2

...

- ViewPager2 là một dạng view sử dụng Fragment
- ViewPager2 cho phép ta dùng thao tác trược tay để chuyển giữa các Fragment. Phối hợp thêm với TabLayout để tạo ra view giống TabHost nhưng mạnh mẽ hơn TabHost



- Bước 1: Tạo Fragment cho mỗi Tab
- Bước 2: Layout chính thêm vào một ViewPager2 và TabLayout

```
<com.google.android.material.tabs.TabLayout
    android:id="@+id/TabLayout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@android:color/transparent"
    app:tabIndicatorColor="@color/black"
    app:tabSelectedTextColor="@color/black" />

<androidx.viewpager2.widget.ViewPager2
    android:id="@+id/ViewPager"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:padding="16dp" />
```

Bước 3: Tạo adapter quản lý ViewPager2

```
public class ViewPager2Fragment extends FragmentStateAdapter {

    1 usage
    public ViewPager2Fragment(@NonNull FragmentActivity fragmentActivity) {
        super(fragmentActivity);
    }

    1 usage
    @NonNull
    @Override
    public Fragment createFragment(int position) {
        if (position == 0) {
            return new Fragment1();
        }
        return new Fragment2();

    }

    @Override
    public int getItemCount() { return 2; }
}
```

❑ Bước 4: Tại Activity setAdapter cho ViewPager2 và setup cho TabLayout

```
ViewPager2Fragment adapterViewPager2 = new ViewPager2Fragment( fragmentActivity: this);
viewPager2.setAdapter(adapterViewPager2);

//Config tablayout
new TabLayoutMediator(tabLayout, viewPager2, new TabLayoutMediator.TabConfigurationStrategy() {
    1 usage
    @Override
    public void onConfigureTab(@NonNull TabLayout.Tab tab, int position) {
        switch (position)
        {
            case 0:
                tab.setText("TabLayout Thứ 1");
                break;
            case 1:
                tab.setText("TabLayout Thứ 2");
                break;
            default:
                break;
        }
    }
}).attach();
```


FPT Education
FPT POLYTECHNIC
Thank you
TIN HỌC VĂN PHÒNG QUỐC TẾ THEO CHUẨN ICDL
Syllabus 5 - Office 2007
LUẬT KINH TẾ
KỸ THUẬT SOẠN THẢO VÀ TRÌNH BÀY VĂN BẢN